Correction to Boeing Document D1-82-0228, "A Fast Procedure for Generating Exponential Random Variables", by M. D. MacLaren, G. Marsaglia, T. A. Bray, dated January 1963:

The array referred to as D(J) in the flow chart, page 5, is tabulated as A_1 on pages 7 and 8.

D1-82-0238

BOEING SCIENTIFIC RESEARCH LABORATORIES

A Fast Procedure for Generating Exponential Random Variables

M. D. MacLaren
G. Marsaglia
T. A. Bray

Mathematics Research

January 1963

D1-82-0228

A FAST PROCEDURE FOR GENERATING EXPONENTIAL RANDOM VARIABLES

by

M. D. MacLaren, G. Marsaglia, T. A. Bray

Mathematical Note No. 283
Mathematics Research Laboratory
BOEING SCIENTIFIC RESEARCH LABORATORIES

January 1963

1. Introduction

In this paper we present a very fast method for generating exponential random variables in a digital computer. The method is exact, in the sense that in theory it returns a random variable with exactly the exponential distribution. In practice the result is an approximation, but the accuracy of the approximation depends only on the word length of the computer. The method is based on the general techniques described in [1] and [2] and the special method for an exponential distribution given in [3].

When programmed for the IBM 7090 computer, the procedure takes 86 microseconds to generate one exponential random variable and requires 590 storage locations. On the IBM 1620 computer the time to generate one random variable is 10.8 milliseconds and 1600 storage locations are used. This computer has a variable word length, and a ten digit number occupies 10 storage locations. The figures given above are for a program having ten digit accuracy. We might remark that multiplication of two ten digit numbers requires 17.8 milliseconds on this computer.

The times quoted above are for standard FORTRAN function subprograms, and include linkages, setting index registers, returning x in normalized floating point, etc.

2. Outline of the procedure.

The procedures for decimal and binary machines are essentially the same. Only the various constants differ. The constants for both cases are listed in Section 4.

We assume that we have available a sequence $u_1, u_2, \ldots$ of independent uniform $[0,1]$ random variables. The problem is to generate an exponential random variable x in terms of the u_i's.

Let f be the exponential density

$$f(t) = e^{-t}, \qquad 0 \leq t.$$

Just as in [2] we write f as a mixture of three densities:

$$f(t) = r_1 g_1(t) + r_2 g_2(t) + r_3 g_3(t).$$

$$r_1 g_1(t) = e^{-(k+1)c} \quad \text{for} \quad 0 \leq kc \leq t < (k+1)c \leq 4.$$

$$r_2 g_2(t) = e^{-t} - e^{-(k+1)c} \quad \text{for} \quad 0 \leq kc \leq t < (k+1)c \leq 4.$$

$$r_3 g_3(t) = e^{-t} \quad \text{for} \quad 4 \leq t.$$

Here k takes only integral values and the constant c is $.1$ for a decimal machine and $.0625$ for a binary machine.

Our procedure for generating x is essentially as follows. Generate a uniform $[0,1]$ random variable u. If $u < r_1$, generate a random variable with density g_1. If $r_1 \leq u < r_1 + r_2$, generate a random variable with density g_2. If $r_1 + r_2 \leq u$, generate a random variable with density g_3.

Examination of the flow chart will reveal, however, that the tests on u do not occur in exactly the above order, and also that they are combined with other tests used to generate the random variables with densities g_1 and g_2.

A random variable with density g_1 may be generated as $y_1 + cu$, where y_1 is discrete, and u is uniform on $[0,1]$. The discrete random variable y_1 assumes values $0,c,2c,\ldots$. It is generated by the method described in [1].

A random variable with density g_2 is produced as $y_2 + c\,\min(v_1,v_2,\ldots,v_z)$, where y_2 is discrete, taking values $0,c,2c,\ldots$. The v_i are independent uniform $[0,1]$, and z is a discrete random variable with the distribution

$$\text{Prob}(z = k) = c^k/[k!(e^c - 1 - c)]$$

$$\text{for} \quad k = 2,3,\ldots.$$

Finally we can generate a random variable with density g_3 as a sum $w + s$, where s has density $(r_1g_1 + r_2g_2)/(r_1 + r_2)$, and w is discrete with the distribution:

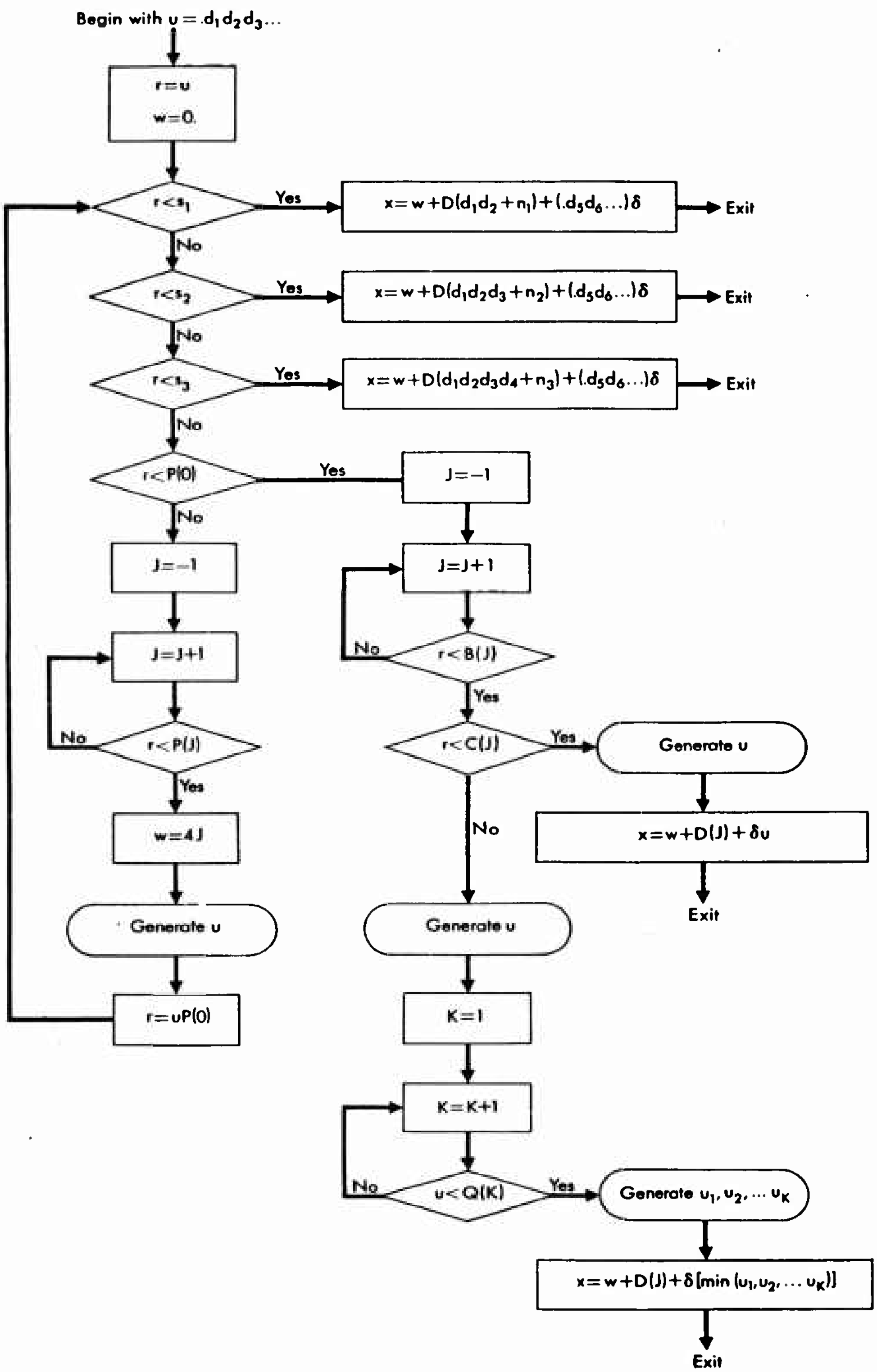
$$\text{Prob}(w = 4k) = e^{-4k}(e^4 - 1), \quad k = 1,2,\ldots.$$

The random variable s is generated by using the procedure to generate the exponential random number x with a slight modification. To be exact we generate an independent uniform $[0,1]$ random variable v and set $u = (r_1 + r_2)v$. The procedure to generate x is re-entered at the first test on u, but now using $u = (r_1 + r_2)v$, which is uniform on $[0, r_1 + r_2]$. This results in the generation of a random variable with density

$$(r_1 g_1 + r_2 g_2)/(r_1 + r_2).$$

3. Flow chart

In this flow chart "generate u" means generate a uniform random variable u. The number r is assumed to have the fractional representation $.d_1 d_2 \ldots$, where the representation is in octal if the program is for a binary space machine. With this notation $d_1 d_2$ is an integer, the representation being in octal for a binary machine. The number δ is 1/10 for a decimal machine and 1/16 for a binary machine.


Begin with u = .d1d2d3...
r=u
w=0.
r<s1
Yes
x=w+D(d1d2+n1)+(.d5d6...)δ
Exit
No
r<s2
Yes
x=w+D(d1d2d3+n2)+(.d5d6...)δ
Exit
No
r<s3
Yes
x=w+D(d1d2d3d4+n3)+(.d5d6...)δ
Exit
No
r<P(0)
Yes
J=−1
No
J=−1
J=J+1
No
r<P(J)
Yes
w=4J
Generate u
r=uP(0)
J=J+1
No
r<B(J)
Yes
r<C(J)
Yes
Generate u
x=w+D(J)+δu
Exit
No
Generate u
K=1
K=K+1
No
u<Q(K)
Yes
Generate u1, u2, ... uK
x=w+D(J)+δ[min (u1,u2, ... uK)]
Exit

4. Constants

For both binary and decimal machines P_k and Q_k are given by the following formulas. The constant c is .1 for a decimal machine and .0625 for a binary machine.

$$(1) \quad P_k = 1 - e^{-4(k+1)} \qquad k = 0,1,\ldots$$

$$(2) \quad Q_k = \sum_{j=2}^{k} c^j/[j!(e^c - 1 - c)] \qquad k = 2,3,\ldots$$

The S_i and n_i are

	Decimal	Binary[1]
S_1	.77	$.45_8$
S_2	.916	$.710_8$
S_3	.9314	$.7427_8$
n_1	198	294
n_2	-764	-144
n_3	-9074	-3639

The B_i and C_i are

	Decimal Machine		Binary Machine	
i	B	C	B	C
0	.9361625820	.9314837418	.94510842161	.94323480082
1	.9393519683	.9362156480	.94628827215	.94528539525
2	.9415112727	.9394089343	.94745642249	.94651430670
3	.9430881445	.9415306939	.94826073606	.94757146958
4	.9442217866	.9431777963	.94894248899	.94843821076
5	.9447661013	.9442476710	.94932953931	.94907597231
6	.9490727662	.9448391766	.94958329340	.94951504666
7	.9529852986	.9491545883	.94980003626	.94967253471
8	.9564834733	.9530173032	.94992008223	.94985597045
9	.9594024969	.9565646369	.95190460176	.95014449802
10	.9620288292	.9594610273	.95361457369	.95196110918
11	.9643851689	.9620617256	.95525892439	.95370563829
12	.9663753874	.9644731130	.95686066511	.95548989495
13	.9681837449	.9664624958	.95839152240	.95693234505
14	.9696461638	.9682369242	.95974907174	.95846135215

[1] The subscript 8 means the number is given in octal representation.

i	Decimal Machine B	Decimal Machine C	Binary Machine B	Binary Machine C
15	.9709809929	.9697058602	.96109651611	.95996010769
16	.9721477967	.9709940089	.96237258589	.96116288541
17	.9731607906	.9722161491	.96344671642	.96237915952
18	.9740454265	.9731906794	.96447806840	.96359303267
19	.9748756955	.9741022884	.96548445255	.96465303841
20	.9756090315	.9749092238	.96632678306	.96554574172
21	.9762878864	.9756546743	.96710374986	.96637002937
22	.9769411563	.9763682022	.96786063548	.96732383419
23	.9774820467	.9770129516	.96853524877	.96788774279
24	.9779150014	.9774905466	.96920558798	.96873186249
25	.9783264218	.9779423592	.96986129425	.96928987221
26	.9787218719	.9784074281	.97048802417	.96987974853
27	.9790899374	.9787424232	.97106734407	.97064928279
28	.9794260891	.9791686442	.97158600948	.97126042295
29	.9797130875	.9795023095	.97208825475	.97180092567
30	.9799999300	.9797154095	.97258093545	.97218820324
31	.9802789666	.9800882467	.97305557876	.97261055489
32	.9805168326	.9802838868	.97352084773	.97317426279
33	.9807267300	.9805541596	.97397651360	.97360757587
34	.9809269259	.9807990826	.97442652331	.97421630891
35	.9811028125	.9809466642	.97485411795	.97458419727
36	.9812764735	.9811351847	.97525425825	.97503048612
37	.9814292281	.9813135507	.97563794313	.97545242949
38	.9815580886	.9814534192	.97601648897	.97577828494
39	.9816843611	.9815896525	.97638555010	.97607969003
40			.97672015684	.97656636068
41			.97705105599	.97695777390
42			.97735451196	.97727717831
43			.97763681248	.97743933438
44			.97791811497	.97781881678
45			.97819899430	.97816010843
46			.97847642598	.97833194781
47			.97874079516	.97856652131
48			.97899467055	.97894151997
49			.97923892352	.97913322108
50			.97947750310	.97941727563
51			.97971601551	.97958029084
52			.97993661346	.97985429235
53			.98015717312	.98004465340
54			.98036825703	.98024848045
55			.98057295172	.98053642199
56			.98077711684	.98073305340
57			.98094419133	.98078047612
58			.98109613615	.98102348810
59			.98124587954	.98119594925
60			.98139402142	.98130639106
61			.98151460394	.98145802551
62			.98163244058	.98159104690
63			.98168436111	.98163745598

For a decimal machine the A_i are:

i	A_i	i	A_i	i	A_i	i	A_i	i	A_i
0	0.0	1	.4	2	.8	3	1.1	4	1.5
5	2.2	6	.1	7	.2	8	.3	9	.5
10	.6	11	.7	12	.9	13	1.0	14	1.2
15	1.3	16	1.4	17	1.6	18	1.7	19	1.8
20	1.9	21	2.0	22	2.1	23	2.3	24	2.4
25	2.5	26	2.7	27	2.6	28	2.9	29	3.1
30	2.8	31	3.2	32	3.0	33	3.3	34	3.6
35	3.4	36	3.5	37	3.7	38	3.8	39	3.9
40	.2	41	.2	42	.2	43	.3	44	.3
45	.3	46	.3	47	.3	48	.3	49	.6
50	.6	51	1.2	52	1.2	53	1.2	54	1.2
55	1.6	56	1.6	57	1.6	58	1.6	59	1.6
60	2.3	61	2.3	62	2.3	63	2.3	64	2.3
65	2.3	66	2.3	67	2.3	68	2.4	69	2.4
70	2.4	71	2.4	72	2.4	73	2.5	74	2.5
75	2.7	76	2.7	77	2.7	78	2.7	79	2.7
80	3.1	81	3.1	82	3.1	83	3.4	84	3.4
85	3.8	86	.5	87	.5	88	.5	89	.6
90	.6	91	.6	92	.6	93	.6	94	.6
95	.7	96	.7	97	.7	98	.9	99	.9
100	.9	101	.9	102	.9	103	1.0	104	1.0
105	1.2	106	1.2	107	1.3	108	1.3	109	1.3
110	1.4	111	1.6	112	1.6	113	1.7	114	1.7
115	1.7	116	1.7	117	1.7	118	1.8	119	1.8
120	1.8	121	1.9	122	1.9	123	2.0	124	2.4
125	2.4	126	2.5	127	2.5	128	2.5	129	2.5
130	2.6	131	2.6	132	2.6	133	2.6	134	2.6
135	2.9	136	2.9	137	2.9	138	2.8	139	2.8
140	2.8	141	2.8	142	3.2	143	3.2	144	3.0
145	3.0	146	3.0	147	3.3	148	3.3	149	3.6
150	3.5	151	3.7	152	.7	153	.7	154	.8
155	.8	156	1.3	157	1.8	158	1.8	159	1.8
160	1.8	161	1.8	162	1.9	163	1.9	164	2.6
165	2.6	166	2.9	167	2.9	168	2.9	169	2.9
170	2.9	171	2.9	172	2.8	173	3.2	174	3.2
175	3.2	176	3.2	177	3.0	178	3.0	179	3.3
180	3.6	181	3.6	182	3.6	183	3.5	184	3.5
185	3.5	186	3.5	187	3.5	188	3.5	189	3.7
190	3.9	191	3.9	192	3.9	193	3.9	194	3.9
195	3.9	196	3.9	197	3.9	198	.0	199	.0
200	.0	201	.0	202	.1	203	.1	204	.1
205	.1	206	.1	207	.1	208	.1	209	.1
210	.4	211	.4	212	.4	213	.4	214	.4
215	.4	216	.5	217	.5	218	.5	219	.5
220	.5	221	.7	222	.7	223	.7	224	.7
225	.8	226	.8	227	.8	228	.8	229	.9
230	.9	231	1.1	232	1.3	233	1.3	234	1.4
235	1.4	236	1.5	237	1.8	238	1.9	239	2.0
240	.0	241	.0	242	.0	243	.0	244	.0
245	.2	246	.2	247	.2	248	.2	249	.2
250	.2	251	.2	252	.3	253	.3	254	.3
255	.3	256	.3	257	.3	258	.6	259	.6
260	.6	261	.6	262	.9	263	1.0	264	1.0
265	1.0	266	1.1	267	1.1	268	1.2	269	1.2
270	1.5	271	1.6	272	1.7	273	2.1	274	2.2

For a binary machine the A_i are:

0	0.	1	0.6250	2	0.6875	3	1.0000	4	1.3125	5	2.0000
6	3.3125	7	2.6875	8	3.3750	9	0.0625	10	0.1250	11	0.1875
12	0.3125	13	0.2500	14	0.3750	15	0.5000	16	0.4375	17	0.5625
18	0.7500	19	0.8125	20	0.8750	21	0.9375	22	1.2500	23	1.0625
24	1.3750	25	1.1875	26	1.1250	27	1.5000	28	1.7500	29	1.8750
30	1.5625	31	1.4375	32	1.6875	33	1.6250	34	2.1875	35	1.9375
36	2.1250	37	2.3125	38	2.0625	39	1.8125	40	2.5000	41	3.0000
42	3.1875	43	2.2500	44	2.9375	45	3.8750	46	2.5625	47	2.3750
48	3.5625	49	2.8750	50	3.4375	51	2.6250	52	3.1250	53	2.8125
54	2.7500	55	3.9375	56	3.7500	57	2.4375	58	3.2500	59	3.6250
60	3.0625	61	3.5000	62	3.8125	63	3.6875	64	0.1250	65	0.1875
66	0.1875	67	0.1875	68	0.1875	69	0.1875	70	0.1875	71	0.3125
72	0.9375	73	0.9375	74	1.2500	75	1.2500	76	1.2500	77	1.1250
78	1.1250	79	1.1250	80	1.1250	81	1.6250	82	2.3125	83	2.3125
84	2.3125	85	2.3125	86	2.0625	87	2.0625	88	1.8125	89	1.8125
90	3.0000	91	2.9375	92	2.9375	93	3.8750	94	3.8750	95	3.8750
96	2.3750	97	2.3750	98	2.3750	99	3.5625	100	3.5625	101	3.5625
102	3.5625	103	3.5625	104	2.8750	105	2.8750	106	2.8750	107	3.4375
108	3.4375	109	3.4375	110	3.4375	111	3.4375	112	3.4375	113	2.8125
114	2.8125	115	2.8125	116	2.8125	117	2.7500	118	2.7500	119	2.7500
120	2.7500	121	2.7500	122	3.9375	123	3.9375	124	3.9375	125	3.7500
126	3.7500	127	3.7500	128	3.7500	129	2.4375	130	2.4375	131	3.6250
132	3.6250	133	3.6250	134	3.6250	135	3.6250	136	3.0625	137	3.5000
138	3.5000	139	3.5000	140	3.5000	141	3.5000	142	3.5000	143	3.8125
144	3.8125	145	3.8125	146	3.8125	147	3.6875	148	3.6875	149	3.6875
150	3.6875	151	3.6875	152	0.1250	153	0.1250	154	0.3125	155	0.3125
156	0.3125	157	0.3125	158	0.3125	159	0.2500	160	0.2500	161	0.3750
162	0.3750	163	0.3750	164	0.3750	165	0.4375	166	0.4375	167	0.8125
168	0.8750	169	0.8750	170	0.8750	171	0.9375	172	0.9375	173	0.9375
174	1.0625	175	1.0625	176	1.3750	177	1.3750	178	1.3750	179	1.1250
180	1.5000	181	1.5000	182	1.5000	183	1.5000	184	1.8750	185	1.8750
186	1.8750	187	1.5625	188	1.6875	189	1.6875	190	1.6875	191	1.6250
192	1.6250	193	1.6250	194	1.6250	195	1.6250	196	2.1875	197	1.9375
198	2.1250	199	2.1250	200	2.1250	201	2.3125	202	2.3125	203	2.0625
204	2.0625	205	2.0625	206	1.8125	207	1.8125	208	1.8125	209	1.8125
210	2.5000	211	2.5000	212	3.0000	213	2.9375	214	2.5625	215	2.3750
216	2.3750	217	2.8750	218	3.1250	219	2.8125	220	2.7500	221	2.4375
222	2.4375	223	3.0625	224	0.	225	0.	226	0.	227	0.0625
228	0.2500	229	0.2500	230	0.2500	231	0.6875	232	0.6875	233	0.6875
234	0.6875	235	0.6875	236	0.6875	237	0.7500	238	0.7500	239	0.7500
240	0.7500	241	0.7500	242	0.8125	243	0.8125	244	0.8125	245	1.0000
246	1.0000	247	1.3750	248	1.3750	249	1.3750	250	1.3750	251	1.5000
252	1.5000	253	1.7500	254	1.7500	255	1.7500	256	1.7500	257	1.7500
258	1.8750	259	1.5625	260	1.5625	261	1.5625	262	1.5625	263	1.5625
264	1.4375	265	1.4375	266	1.4375	267	1.4375	268	1.4375	269	1.4375
270	1.4375	271	1.6875	272	1.6875	273	2.1875	274	2.1875	275	1.9375
276	1.9375	277	1.9375	278	2.0000	279	2.0000	280	2.0000	281	2.0000
282	3.1875	283	2.2500	284	2.2500	285	2.2500	286	2.5625	287	3.3125
288	2.6250	289	2.6250	290	2.6875	291	2.6875	292	3.2500	293	3.3750
294	0.	295	0.	296	0.	297	0.0625	298	0.0625	299	0.0625
300	0.2500	301	0.2500	302	0.5000	303	0.5000	304	0.4375	305	0.6875
306	0.5625	307	0.7500	308	0.8125	309	0.8750	310	1.0000	311	1.1875
312	0.1250	313	0.1250	314	0.1250	315	0.1875	316	0.1875	317	0.1875
318	0.3125	319	0.3125	320	0.3750	321	0.3750	322	0.4375	323	0.6250
324	0.6250	325	0.5625	326	0.9375	327	1.2500	328	1.3125	329	1.0625
330	1.1250										

REFERENCES

[1] G. Marsaglia, Generating Discrete Random Variables in a Computer, Comm. Assoc. Comp. Mach., Vol.6, No. 1 (1963).

[2] G. Marsaglia, M. D. MacLaren, T. A. Bray, A Fast Procedure for Generating Normal Random Variables, to be published .

[3] G. Marsaglia, Generating Exponential Random Variables, Ann. Math. Stat., Vol. 32, p. 899, (1961)